UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/998,756	12/03/2001	Jeff L. Hunter	TI-33109	6454

23494 7590 05/20/2005

TEXAS INSTRUMENTS INCORPORATED
P O BOX 655474, M/S 3999
DALLAS, TX 75265

EXAMINER	
RUTTEN, JAMES D	

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 05/20/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Examiner-Initiated Interview Summary	Application No.	Applicant(s)	
	09/998,756	HUNTER ET AL.	
	Examiner	Art Unit	
	J. Derek Rutten	2192	

All Participants:

(1) *J. Derek Rutten*.

(2) *Robert Marshall, Reg. 28,527*.

Status of Application: *Not Allowable*

(3) _____.

(4) _____.

Date of Interview: *10 May 2005*

Time: _____

Type of Interview:
- ☒ Telephonic
- ☐ Video Conference
- ☐ Personal (Copy given to: ☐ Applicant ☐ Applicant's representative)

Exhibit Shown or Demonstrated: ☒ Yes ☐ No

If Yes, provide a brief description: *Draft amendment to claims 1-3, 17-19, and 21-23 attached.*

Part I.

Rejection(s) discussed:

Claim 3 under 35 USC 103(a) over Falik.

Claims discussed:

1-3

Prior art documents discussed:

None

Part II.

SUBSTANCE OF INTERVIEW DESCRIBING THE GENERAL NATURE OF WHAT WAS DISCUSSED:

See Continuation Sheet

Part III.

☐ It is not necessary for applicant to provide a separate record of the substance of the interview, since the interview directly resulted in the allowance of the application. The examiner will provide a written summary of the substance of the interview in the Notice of Allowability.

☒ It is not necessary for applicant to provide a separate record of the substance of the interview, since the interview did not result in resolution of all issues. A brief summary by the examiner appears in Part II above.

TUAN DAM
SUPERVISORY PATENT EXAMINER

(Examiner/SPE Signature)

(Applicant/Applicant's Representative Signature – if appropriate)

Continuation of Substance of Interview including description of the general nature of what was discussed: Examiner Rutten contacted Mr. Marshall on May 6, 2005, to suggest an examiner's amendment to the independent claims in light of potential allowable subject matter regarding searching a software memory map. And if such amendments were agreeable, claims 14-16 should be amended back to their original recitation of "the software memory map". Mr. Marshall agreed to consider such an amendment. On May 10, 2005, Mr. Marshall left a voice mail message indicating that the applicant did not agree with the suggested amendment, and would prefer a new action. Mr. Marshall further indicated in the message that claims 14-16 should contain recitations of a "software memory map" similar to claim 2. .

Allowable Subject Matter

1. The following claims 1-3, drafted by the examiner and considered to distinguish patentably over the art of record in this application, are presented to applicant for consideration:

1. (CURRENTLY AMENDED) A method for maintaining coherency of software breakpoints in shared memory when debugging a multiple processor system, the method comprising the steps of:

activating a first debug session associated with a first processor of a plurality of processors and at least a second debug session associated with a second processor of the plurality of processors;

creating a software memory map of the memory usage of a plurality of processors in the system to be debugged;

searching the software memory map to find a first plurality of processors having read access to a shared memory location;

updating a software representation maintained for software breakpoints for each of the first plurality of processors; and

setting a first software breakpoint in a shared memory location in the first debug session by writing the software breakpoint instruction in the shared memory location such that all debug sessions are notified of the setting of the breakpoint; and

clearing the first software breakpoint in the shared memory location in the second debug session such that all debug sessions are notified of the clearing of the breakpoint.

2-3. (CANCELED)

17. (CURRENTLY AMENDED) A software development system, comprising:

a memory storage system holding a software development tool program;

a host computer connected to the memory storage system, the host computer operable to execute the software development tool program;

a test port for connecting to a target hardware systems the hardware system being comprised of multiple processors with common shared memory and operable to execute an application program; and

wherein the software development tool is operable to support debugging of the application program executing on the target hardware system using a method for maintaining coherency of software breakpoints in shared memory when debugging a multiple processor system, the method comprising the steps of:

activating a first debug session associated with a first processor of a plurality of processors and at least a second debug session associated with a second processor of the plurality of processors;

creating a software memory map of the memory usage of a plurality of processors in the system to be debugged;

searching the software memory map to find a first plurality of processors having read access to a shared memory location;

updating a software representation maintained for software breakpoints for each of the first plurality of processors; and

setting a first software breakpoint in a shared memory location in the first debug session by writing the software breakpoint instruction in the shared memory location such that all debug sessions are notified of the setting of the breakpoint; and

clearing the first software breakpoint in the shared memory location in the second debug session such that all debug sessions are notified of the clearing of the breakpoint.

18-19. (CANCELED)

21. (CURRENTLY AMENDED) A digital system, comprising:

multiple processors with common shared memory for executing an application program; and

wherein the application program was developed with a software development system using a method for maintaining coherency of software breakpoints in shared memory when debugging a multiple processor system, the method comprising the steps of:

activating a first debug session associated with a first processor of a plurality of processors and at least a second debug session associated with a second processor of the plurality of processors;

creating a software memory map of the memory usage of a plurality of processors in the system to be debugged;

searching the software memory map to find a first plurality of processors having read access to a shared memory location;

updating a software representation maintained for software breakpoints for each of the first plurality of processors; and

setting a first software breakpoint in a shared memory location in the first debug session by writing the software breakpoint instruction in the shared memory location such that all debug sessions are notified of the setting of the breakpoint; and

clearing the first software breakpoint in the shared memory location in the second debug session such that all debug sessions are notified of the clearing of the breakpoint.

22-23. (CANCELED)